Synthesis of 3D Sound Movement by Embedded DSP

Shinya Komata[1], Noriaki Sakamoto[1], Wataru Kobayashi[1,2],
Takao Onoye[1] and Isao Shirakawa[1]
[1]Dept. of Information Systems Engineering
Graduate School of Information Science and Technology, Osaka University
2-1 Yamada-Oka, Suita, Osaka, 565-0871 Japan
Phone: +81-6-6879-7808, Fax: +81-6-6875-5902
E-mail: {shinya, sakamoto, kobachan}@ise.eng.osaka-u.ac.jp, {onoye, sirakawa}@ist.osaka-u.ac.jp
[2]Arnis Sound Technologies, Co., Ltd.
2-7-9 Kita-Senzoku, Ota, Tokyo, 145-0062 Japan
Phone: +81-3-5749-3621, Fax: +81-3-5749-3625, E-mail: kobachan@arns.com

Abstract: A single DSP implementation of 3D sound movement is described. With the use of a realtime 3D acoustic image localization algorithm, an efficient approach is devised for synthesizing the 3D sound movement by interpolating only two parameters of "delay" and "gain". Based on this algorithm, the realtime 3D sound synthesis is performed by a commercially available 16-bit fixed-point DSP with computational labor of 65 MIPS and memory space of 9.6k words, which demonstrates that the algorithm can be used even for the mobile applications.

1. Introduction

Recently a number of 3D sound synthesis algorithms have been investigated with the use of head-related transfer functions (HRTFs)[1][2][3], where the overall characteristics of HRTFs are attempted to be realized throughout the audible frequency band, and hence the process of approximating precisely the whole HRTF behaviors involves too intensive computational complexity for a single DSP to perform the realtime 3D sound synthesis.

To cope with this technical difficulty, a realtime algorithm of the 3D sound localization is devised dedicatedly for embedded applications to be implemented by a single DSP[4]. This algorithm is distinctive in that the function form of a given HRTF is divided into three portions of low, intermediate, and high frequency subbands, such that each of the divided portions can be approximated efficiently by employing a distinct optimal filtering structure so as for a single DSP to perform the 3D sound synthesis dedicatedly for mobile computing.

The computational complexity of the algorithm can be reduced to such an extent that even moving 3D sound images can be synthesized in realtime by a commercially available DSP. However, due to the overhead of updating filter parameters necessary for this algorithm, there still remain a number of technical issues for the synthesis of the realtime sound movement.

Motivated by this practical necessity, the present paper devises a sophisticated approach to the synthesis of 3D sound movements.

2. Synthesis of 3D Sound Localization

As the preliminary of 3D sound movement, the synthesis algorithm[4][5] of 3D sound localization is briefed.

This algorithm is distinctive in that the audible frequency band of a given HRTF is divided into three subbands, such that the 3D sound synthesis in each subband can be efficiently achieved by a distinct optimal filtering structure. In comparison with conventional methods, this algorithm can reduce a great amount of computational complexity so that a single DSP implementation can be achieved.

Figure 1 shows an outline of the implementation process of this 3D sound synthesis algorithm. First, at the

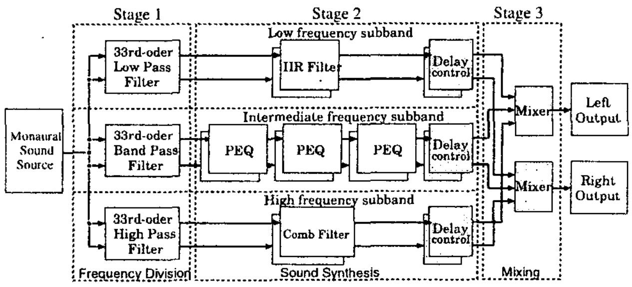

Figure 1. Block diagram of 3D sound synthesis algorithm.

frequency division stage, a given monaural sound source is divided into three frequency subbands; low, intermediate, and high frequency subbands; with the use of the 33rd-order low pass, band pass, and high pass FIR filters, respectively.

Second, at the sound synthesis stage, the 3D sound synthesis is performed by approximating HRTF behaviors with the use of a first-order Shelving IIR filter in the low subband, three parametric equalizers (PEQs) constructed by second-order IIR filters in the intermediate subband, and a comb filter in the high subband, respectively. After this process, the delay differences can be controlled by delay control buffers. The use of "delay" is intended mainly for making the arrival time difference between left and right ears and for concealing filter processing delay differences.

Finally, audio data in the left and right channels in three subbands are mixed with the use of "gain" parameters. This "gain" is for controlling sound levels by amplifying audio data in the left and right channels. Through this series of filter processing, the synthesis of 3D sound localization can be achieved.

3. Interpolation Scheme for DSP Implementation

To enhance the 3D sound effects, the functionalities of not only sound localization but also sound movement are indispensable. Thus, in addition to the above algorithm, a new scheme of synthesizing 3D sound movement is devised, which is to perform *smooth* and *realistic* 3D sound movements in realtime with the use of a single DSP.

Given a pair of sound movement route and time, generally there are three methods to synthesize 3D sound movements as follows. One is to prepare a number of sound image positions, whose HRTFs are measured by binaural recording in advance. Another is to generate a 3D sound movement between two sound image positions by interpolating localized sounds of these positions. The other is to reproduce the 3D sound at each position in a movement, by interpolating filter parameters of two sound image positions.

First, we discuss the method to prepare parameters for a number of sound image positions. In the 3D sound synthesis algorithm, the memory size of filter parameters per sound image position is 321 words, which is small enough for embedded processing in case of localizing a sound image in a fixed position. However, the more smooth is the 3D sound movement, the larger is the number of sound image positions, which requires a great amount of memory space. Hence this method suffers from impractical memory space against the single DSP implementation.

In the case of the interpolation between two localized sounds, 3D sound movement effects can be easily achieved by mixing these localized sounds. However, noting that both of the sound levels at these two positions are necessary in realtime for all sampling points in the course of sound movement, it can be verified that the total computational labor is more than twice of that of the 3D sound localization. As a result, this method also has the difficulty of the realtime processing by a single DSP.

As for the filter parameter interpolation, a trivial scheme is to calculate all filter parameters by interpolating two HRTFs at each sampling point on the route. However, this scheme is impractical for realtime applications from the aspects of computational labor and memory size.

To lessen the computational labor as well as the memory size, our approach is to interpolate "delay" parameters at the sound synthesis stage and "gain" parameters at the mixing stage, as shown in Figure 1, both of which affect strongly the human auditory sense of sound movement. The number of delay parameters and those of gain parameters are both six, since these parameters are set for three subbands of left and right channels as shown in Figure 1.

As to the interpolation, there are linear and spline interpolations. According to [6], which compares the accuracy between these two interpolations for a given number of HRTFs, the accuracy difference between linear and spline interpolations is negligibly small in case of setting twenty four HRTFs in a horizontal plane. Therefore, we adopt the linear interpolation in order to reduce the computational labor.

Henceforth an effective interpolation scheme of two HRTFs is discussed for the sound movement between two positions by using "delay" and "gain" parameters.

The interpolation scheme is briefed as follows.

Assume that a sound image is moving from position "A" to position "B" along a circle in a moving time T. First, calculate delay/gain differences $d_i (i = 1, 2, ...)$ between two positions in three subbands. Then, for each d_i let a time interval t_i be T/d_i.

For example, as illustrated in Figure 2, assume that the *i-th* parameter is the gain, and their values at "A" and "B" are "5" and "15", respectively. Then the difference d_i is "$15 - 5 = 10$", and the time interval t_i is $T/10$. Thus in the course of the sound movement from "A" to "B", the parameter changes as "5,6,....14,15" in each time interval. On the contrary, if the sound image moves from "B" to "A", then the parameter decreases as "15,14,....6,5" in each time interval. In this manner, the linear interpolation for delays/gains can be executed.

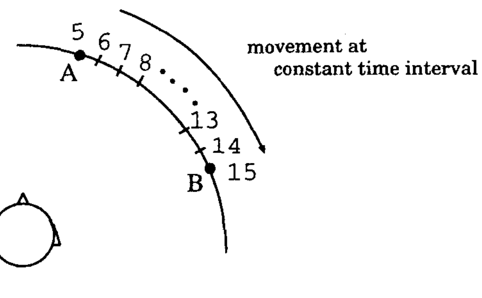

Figure 2. Example of movement between concentric positions A and B.

As for the filter parameters other than the delay/gain in the sound movement from "A" to "B", we keep parameter values at the starting position "A" until the sound image arrives at the destination position "B", to renew them when the sound image arrives at "B", whereas in case of the movement from "B" to "A", the parameter values at "B" are used until the sound image arrives at the position "A".

To enable a single DSP implementation of this scheme, reference tables are used to save computational costs of calculating T/d_i, since the division by a DSP needs a number of instructions. This scheme can also reduce the memory size for storing intermediate data. Given a moving time T and delay/gain differences d_i, the quotient of T/d_i can be derived directly from these tables.

The size of quotient table depends on the precision of d_i. For example, if we prepare tables for 16-bit, 12-bit, and 8-bit data, 64K, 4K, and 256 words memory, respectively, are necessary for each T. However, in embedded applications the size of the table must be kept small. Although the delay and gain parameters are represented in the 16-bit precision in our 3D sound localization, where leading two bits are always 0's for all delays and gains in our case, 6 bits are extracted as illustrated in Figure 3 to refer the quotient table, each with 64 words.

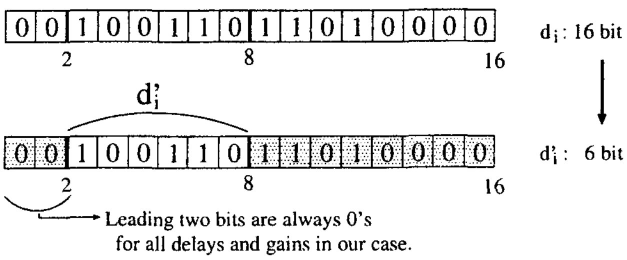

Figure 3. Extraction of d_i's 6 bits.

As for the number of T's, it must be optimized according to the DSP memory size. For example, if we prepare eight T's, i.e. moving speed in 8 levels, totally 512 words memory is necessary for quotient table, as exemplified in Figure 4.

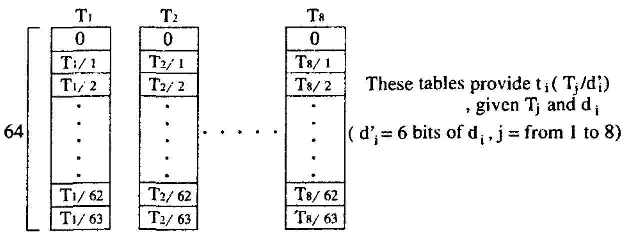

Figure 4. Quotient table.

As a result, this approach reduces the computational labor by 50 MIPS, in comparison with the conventional division-based calculation.

4. Implementation Results

In addition to the block diagram shown in Figure 1, the newly introduced processes of this interpolation scheme are to calculate "delay" and "gain" differences d_i and time intervals T/d_i, and to modify delay and gain parameters in every time interval. Thus the new block diagram for the synthesis of 3D sound movement is drawn as shown in Figure 5.

Table 1 summarizes the computational costs with the use of a Texas Instruments DSP TMS320C54x, where it should be pointed out that totally 65.7 MIPS is necessary. This result shows that the total computational cost is quite smaller than the maximum performance of the DSP.

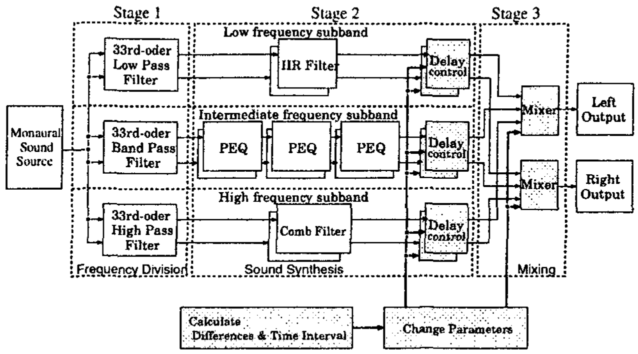

Figure 5. Block diagram of 3D sound synthesis algorithm for 3D sound movement.

Table 1. Computational costs.(unit: MIPS)

	Frequency division	Sound synthesis	Mixing	Additional process
Low subband	5.71	4.80	4.90	14.1
Intermediate subband	9.55	15.4		
High subband	5.04	6.24		
Total	65.7			

Figure 6 shows a memory map for the 3D sound synthesis. Program and filter parameters are assigned to ROM, while the data buffer and the data area to RAM. The memory size necessary for filter parameters per sound image position is 321 words.

For example, in the case of implementing the 3D sound movement for 24 sound image positions with angles of every 15 degrees, only 9.6k and 1.4k words are assigned to ROM and RAM, respectively, which are much less than those of a single DSP TMS320C5409 (16k words ROM, 32k words RAM).

The power consumption of this DSP implementation of the 3D sound synthesis is evaluated as 108 mW, which is based on 0.9 mA/MIPS of the DSP core consumption in FIR processing, the dominant part in this implementation, with 1.8 V supply voltage (due to the datasheet

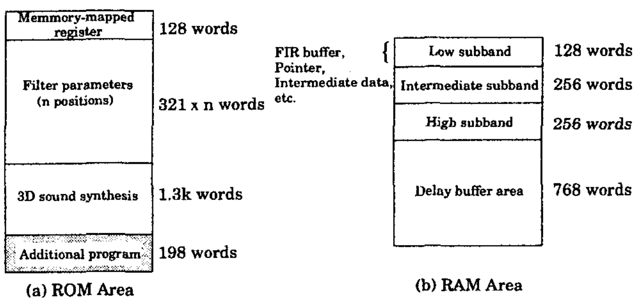

Figure 6. Memory map of synthesizing 3D sound movement.

of TMS320C54x[7][8]). Since the power consumption of a commercially available portable game system is 600 mW, it can be seen that the addition of this 3D sound synthesis system becomes 17 % increase. Considering that a portable game system generally uses a specific chip, an ASIC implementation instead of a DSP one enables much more reduction of power consumption.

Consequently, this algorithm can synthesize the 3D sound movement in realtime by employing a single DSP, and provides us with comfortable listening environments for the 3D sound effects.

5. Conclusion

The present paper has described an effective approach to the synthesis of the 3D sound movement, which adopts the effective interpolation scheme of two HRTFs at concyclic positions, by using only a few parameters which affect strongly the human auditory sense of sound movement. As a result, by employing a Texas Instruments DSP, TMS320C54x, the realtime 3D sound synthesis can be performed at low computational labor of 65.7 MIPS, with small memory area of 9.6k words and low power consumption of 108 mW at 1.8 V supply voltage. The implementation results demonstrate that the realtime 3D sound movement can be performed in mobile applications with the use of an embedded DSP.

Development is continuing on the enhancement of more sophisticated 3D sound synthesis mechanisms which can handle smooth sound movement and expansion.

References

[1] C. P. Brown and R. O. Duda, "A structural model for binaural sound," *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 476–488, Sept. 1998.

[2] S. Lee, K. Lee, D. Ryu, and K. Park, "High performance real time 3-D audio engine,"*in Proc. Int. Technical Conf. Circuits/Systems, Computers and Communications*, vol. 2, pp. 1136-1139, June 2001.

[3] M. Okamoto, I. Kinoshita, S. Aoki, and H. Matsui, "Sound image rendering system for headphones,"*IEEE Trans. Consumer Electronics*, vol. 43, no. 3, pp. 689-693, Aug. 1997.

[4] W. Kobayashi, N. Sakamoto, T. Onoye, and I. Shirakawa, "3D acoustic image localization algorithm by embedded DSP," *IEICE Trans. Fundamentals*, vol. E84-A, no. 6, pp. 1423–1430, June 2001.

[5] N. Sakamoto, W. Kobayashi, T. Onoye, and I. Shirakawa: "DSP implementation of 3D sound localization algorithm for monaural sound source," in *Proc. The 8th IEEE International Conference on Electronics, Circuits and Systems*, Malta, pp. 1061–1064, Sept. 2001.

[6] T. Nishino, S. Kajita, K. Takeda and F. Itakura, "Interpolation of the Head Related Transfer function on the horizontal plane," *J. Acous. Soc. Jpn.*, no. 55, pp. 91–99, 1999(in Japanese)

[7] Texas Instruments Inc.,"Calculation of TMS320LC54x Power Dissipation, Application Report," June 1997.

[8] Texas Instruments Inc.,"TMS320VC5402 Datasheet," Aug. 2000.